Arnol Johnson

LINEAR PROGRAMMING

Library
U. S. Naval Postgraduate School
Monterey, California

LINEAR PROGRAMMING

U. S. Naval Postgraduate School
Monterey, California

ACKNOWLEDGEMENTS

I wish to express my appreciation to Professor James H. Greene for initiating my interest in this subject and providing assistance, and to Professor Abraham Charnes for the use of his notes on linear programming.

TABLE OF CONTENTS

	Page
INTRODUCTION	1
MATHEMATICS OF LINEAR PROGRAMMING	8
Notation	8
Matrices	10
Vectors and Points in Space	13
Convex Sets	18
Basis	19
Slack Vectors	20
GENERAL FORM OF LINEAR PROGRAMMING	21
Simplex Method	21
Example of Simplex Method	23
TRANSPORTATION METHOD	43
Example of Transportation Method	46
Example of Transportation Method, Fictitious Route	51
ASSIGNMENT METHOD	55
Example of Assignment Method	55
DUAL PROBLEM	59
SUMMARY	60
REFERENCES	62

INTRODUCTION

A recent addition to the tools of management and production engineering is the technique of linear programming. The purpose of this project, a result of a survey of literature in this field, is to summarize the fundamentals of linear programming. Toward that objective, the following phases are presented: a summary of the mathematical concepts involved, three methods of solution and their characteristics, and the significance of the dual theorem. The general form, the simplex method, and two of its special cases, the transportation method and the assignment method are presented. Two basic references for the mathematics of linear programming are references (1) and (4).

The names of the special cases should not be taken literally. For example, the transportation method may be used for problems with no physical transportation involved. Similarly, the assignment method may be used in quality control, purchasing, set up time, and such other diverse problems. In the development of the special cases, the type of problem was essentially the same as the name implied, but in recent years the methods have been applied to a variety of problems.

The general form, the simplex method, was developed by Dantzig.[2] Most of the work in the field of linear programming has been conducted by economists, mathematicians, and consulting firms. However, the number of qualified personnel for this type of work is limited compared to the needs of industry. The usefulness of mathematical programming depends to a large extent on the ingenuity of the personnel formulating the mathematical model. Many management and technical publications contain reports on the use of this type of programming but are necessarily brief due to publication requirements and other factors.

It should be noted that many of the calculations involved in linear programming can be handled in a routine fashion utilizing desk calculators, but a large scale problem is better solved by computers. Regarding the use of desk calculators, one of the advantages of linear programming is that the mechanics of solution are simple and may be handled by lesser skilled personnel to obtain the best solution to a given problem. This characteristic should be borne in mind as an aid in freeing engineering and management personnel for more demanding work commensurate with their training. The most important phase of the programming solution is that of setting up the initial problem. Once this has been done, the mechanics can be performed by clerical personnel.

Operations research, operations analysis, and mathematical programming have received wide publicity since World War II. The majority of the research and application was conducted during World War II, and management is now in the process of utilizing some of the methods in industry. There is an abundance of literature concerning the above topics and no attempt will be made to describe or discuss those fields. Essentially, linear programming is a method or technique within the field of mathematical programming, and attention will be limited to that phase.

Linear programming is a systematic mathematical approach to a given problem to arrive at an optimal or best solution. It involves an objective and a set of restrictions or constraints. The procedure has been effectively used in industrial problems where a large number of alternative choices were available. From case histories published, it may be said that the usefulness of the method is proportional to the size and complexity of the situation. Until more research has been

conducted, some industries may not be able to use linear programming to its full capacity. However, by examining a local problem for application and trying to formulate the problem by the methods of linear programming, possible cost and time savings may be realized.

One of the limitations of the method is the requirement of linearity of the variables involved. Due to this restriction, all variables such as time, cost, and quantity must be of the first order. That is, the ratio of input to output must be a fixed ratio and independent of the level of production or process involved. For example, if one unit of input results in one unit of output then ten units of input must result in ten units of output. Dorfman[3] has summarized the basic assumptions of linear programming as follows:

1. The productive opportunities of an economy or economic unit are defined by the resources and the productive processes available to it. The quantities of at least some of the resources are finite and so is the number of productive processes available.

2. Any productive process may be used at any positive level consistent with the supply of resources available. The consumption of resources and the output of products is proportional to the level at which the process is used.

3. Several productive processes may be used simultaneously if the supply of resources is adequate. If this is done, the consumptions of the individual processes used, and the output of products is the sum of the outputs of the individual processes.

Dorfman[3] further reduces the three assumptions to the following postulates:

a. Linearity. By definition, in linear programming, each process is characterized by certain ratios of the quantities of the inputs to each other and to the quantities of each of the outputs. These ratios are defined to be constant and independent of the extent to which the process is used.

b. Divisibility. It is assumed that any process can be used to any positive extent so long as sufficient resources are available; indivisibilities and "lumpiness" in production are ignored.

c. Additivity. It is assumed that two or more processes can be used simultaneously, within the limitations of available resources, and that if this is done the quantities of the outputs and inputs will be the sums of the quantities which would result if the several processes were used individually.

d. Finiteness. It is assumed that the number of processes available is finite.

The terms: resources, products, and production processes have very broad meanings as used above and are not limited to the concepts often used in industry.

Programming in this sense may be considered as arriving at the solution systematically; the term stems from the "activity analysis" approach of reference (4). The objective of a problem may be to minimize or maximize some function such as time, cost, or material. The constraints may be set by machine time available, raw materials, capital to be invested, time in transportation, quantity to be produced, or other restrictions.

Initially, the particular problem is formulated mathematically into a set of inequalities with more variables present than the number of equations. If it is possible to set up as many equations as variables present, then the obvious method is a solution by simultaneous equations. However, in such an approach negative answers may result, and this situation may also be handled by linear programming. A condition warranting the application of linear programming is that there exist a large number of alternative choices in which any one choice of a subordinate phase will affect the outcome of the entire plan. If there is adequate machine time available in a plant for a particular product, there is still an advantage in using the method. However, if

a number of products can be processed on any number of machines in a plant, a trial and error procedure is far too time consuming, and even then the solution may not be the best one. The word scheduling is purposely omitted here for if the problem involves a decision on sequence of operations, present methods of linear programming cannot handle this completely. Research is being conducted along this line such as noted in reference (5).

Charnes[6] has described a linear programming problem as follows: If a problem can be written in the following form,

Given a criterion of choice

(1) $\sum_{j=1}^{n} c_j x_j$ which is to be minimized or maximized, and is subject to the two types of constraints or conditions that follow:

(2) $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$, $\quad i = 1, 2, 3....m$, and

(3) $x_j \geq 0$, $\quad j = 1, 2, 3....n$; then the problem is a linear programming problem.

Here, the objective is the criterion of choice (1), and the constraints are contained in (2) and (3). For maximizing, the following meanings may be associated with the variables:

x_j = a variable to be determined such as the quantity of a part or a product.

c_j = profit per unit.

a_{ij} = a capacity constraint such as a unit production time.

b_i = a stipulation such as production capacity expressed in time available.

Although solution to the following problem is obvious, it is presented to show formulation to the above conditions. Suppose there are two products P_1 and P_2, one machine available that can produce either product, and the available machine time is 8 hours.

P_1	P_2
x_1 = units to produce	x_2 = units to produce
2 hrs = production time per unit	4 hrs = production time per unit
\$4.00 = unit profit	\$2.00 = unit profit

Problem:

(1) $\$4.00\ x_1 + \$2.00\ x_2$, maximize profit

(2) $2\text{ hrs}\ x_1 + 3\text{ hrs}\ x \leq 8\text{ hrs}$

$x_1 \geq 0$, and $x_2 \geq 0$

Aside from product demands or promised delivery, the answer is apparent by inspection. However, as the number of products, machines, or production paths available increase and product demands are considered, the problem becomes proportionally complex. The linear programming method is carried out by successive steps, and if a solution is possible, a maximum profit case will be reached. Also, lower profit cases can be considered for factors such as balancing machine time or other economic reasons.

Regarding the construction of the mathematical model, Bellman[7] has described operations of a multi-stage process to be of two types: deterministic and stochastic. Deterministic may be considered as known data, the solution being completely determined for each choice of alternative, whereas stochastic data involves a probability distribution of some form. References (27) and (28) discuss methods where probability

considerations are given both to the functional and to the constraints involved. Drawing an analogy in the field of mechanics concerning the use of statics or dynamics, there is as large a field of application in Industrial Engineering for use of the deterministic concept as of the stochastic concept.

Many examples of linear programming emphasize a solution for the number of products to produce, ship, buy, or similar functions. However, one of the most useful attributes of a linear programming solution is that of cost analysis and the effect of changing constraints or stipulations such as minimum or maximum requirements, unit time, unit cost, unit rates, and similar factors.

MATHEMATICS OF LINEAR PROGRAMMING

This section is a summary of the basic mathematical tools and concepts involved in linear programming. It is not all inclusive but contains the essential features normally encountered in the literature concerning this field. References should be consulted for precise definitions and details.

MacDuffe[8] points out that the importance of matrix theory derives from the rules of combination of matrices and the fact that they may be represented as square arrays is incidental. The term "tableau," as used in linear programming for the tabular array, indicates the array to be a member of matric algebra. The tableau is also referred to as a frame, since it describes the convex hull of the polyhedral cone by means of half lines spanning the cone.

Notation

Summations.

A series of values, known or unknown, may be given as:

$$a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + a_5x_5 \cdots + a_nx_n$$

Or more compactly:

$$\sum_{j=1}^{n} a_jx_j \qquad j = 1, 2, 3, \ldots n$$

For a set of expressions:

$$a_{11}x_1 + a_{12}x_2 \cdots + a_{1j}x_j \cdots + a_{1n}x_n = \sum_{j=1}^{n} a_{ij}x_j \qquad i = 1$$

$$a_{21}x_1 + a_{22}x_2 \cdots + a_{2j}x_j \cdots + a_{3n}x_n = \sum_{j=1}^{n} a_{ij}x_j \qquad i = 2$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$a_{i1}x_1 + a_{i2}x_2 \cdots + a_{ij}x_j \cdots + a_{in}x_n = \sum_{j=1}^{n} a_{ij}x_j \qquad i = \text{row number}$$

$$a_{m1}x_1 + a_{m2}x_2 \cdots + a_{mj}x_j \cdots + a_{mn}x_n = \sum_{j=1}^{n} a_{ij}x_j \qquad i = m$$

The matrix of this set may be denoted by $[a_{ij}]$ or A, with each element a_{ij} corresponding to the element in the i^{th} row and j^{th} column.

If the column values of each x_j are not identical, the above may be extended to use of a double summation:

$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_{ij}$ Where the corresponding elements, a_{ij} and x_{ij}, are multiplied and summed for total value.

In matrix notation the set of expressions may be shown as:

$$[a_{ij}] \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_j \end{bmatrix} = \sum_{j=1}^{n} a_{ij} x_j \qquad i = 1, 2, 3 \ldots m$$

Inequalities and Equalities.

Symbols used:

$y > x$	y is greater than x
$y \geqslant x$	y is greater than or equal to x
$x < y$	x is less than y
$x \leqslant y$	x is less than or equal to y

These symbols represent such restrictions as "produce no more than," "stock no less than," "pay no more than," "produce at least," etc. In the above examples the latter two were obtained by reversing the symbol and order of variables. However, in solving a problem containing a set of mixed inequalities, it is necessary to make the system consistent. This is carried out by multiplying through by minus one, which reverses the inequality and preserves the order of variables.

Example: $x_1 + x_2 + x_3 \geqslant y$

$x_1 + x_2 + x_3 \leqslant z$

Depending on the problem being solved (maximize or minimize), either

inequality may be multiplied through by minus one, giving:

$$-x_1 - x_2 - x_3 \leqslant -y \qquad \text{or} \qquad x_1 + x_2 + x_3 \geqslant y$$
$$x_1 + x_2 + x_3 \leqslant z \qquad\qquad -x_1 - x_2 - x_3 \geqslant -z$$

Matrices

Essentially, matrices are sets of numbers, real or complex, arranged in rectangular arrays and which obey certain rules of multiplication.

Matrix. A rectangular array of elements arranged in m rows and n columns. The matrix is of order (m, n). It is written as a (mxn) matrix. If $m = n$, it is termed a square matrix of order n.

Example:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & a_{m3} \cdots & a_{mn} \end{bmatrix} = [a_{ij}] \qquad \begin{matrix} i = 1, 2 \ldots . m \\ j = 1, 2 \ldots . n \end{matrix}$$

Where a_{ij} is each element.

Diagonal Matrix. A square matrix having all off diagonal elements equal to zero. The diagonal elements may equal any value except zero.

Example:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Unit Matrix. A diagonal matrix having diagonal elements equal to unity and off diagonal elements equal to zero.

Example:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This matrix is also symmetric because the matrix is equal to its transpose.

Transpose of a Matrix. The transpose of a matrix is obtained by interchanging rows and columns and is denoted by a prime.

Example: $A = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 2 & 1 \\ 2 & 3 & 3 \end{bmatrix}$, Transpose $= \begin{bmatrix} 1 & 3 & 2 \\ 2 & 2 & 3 \\ 1 & 1 & 3 \end{bmatrix} = A'$

Row Vector. A set of n numbers in a row, a matrix of order (1, n).

$$[a] = [a_1 , a_2 , a_3 \cdots a_n]$$

Column Vector. A set of m numbers in a column, a matrix of order (m, 1).

$$[b] = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdot \\ \cdot \\ b_m \end{bmatrix}$$

Addition. To add, both matrices must have the same number of rows and columns, corresponding elements are added.

Subtraction. The same rule applies as for addition with corresponding elements subtracted.

Multiplication of Two Matrices. Matrix multiplication is defined such that linear transformations may be concisely and conveniently expressed. Two matrices can be multiplied only if the number of rows in the first matrix is equal to the number of columns in the second matrix. If this condition is satisfied the matrices are said to be conformable. The result of this multiplication is a matrix with the number of rows equal to the number of rows in the first matrix, and the number of columns equal to the number of columns in the second matrix. Elements of the product matrix are obtained by multiplying a row of the first matrix by the same numbered column of the second matrix.

Examples:
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (a_{11}x + a_{12}y + a_{13}z) \\ (a_{21}x + a_{22}y + a_{23}z) \\ (a_{31}x + a_{32}y + a_{33}z) \end{bmatrix}$$

$$[x\ y\ z] \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} =$$

$$= [(xa_{11} + ya_{21} + za_{31})(xa_{12} + ya_{22} + za_{32})(xa_{13} + ya_{23} + za_{33})]$$

Multiplication of a row vector by a column vector results in a scalar,

$$[a_1\ a_2\ a_3] \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = (a_1b_1 + a_2b_2 + a_3b_3)$$

Multiplication of a Matrix by a Scalar. Each element of the matrix is multiplied by the scalar.

Division by a Matrix. Division by a matrix is performed by multiplication of its inverse.

Example: $A\ B = C$

$$B = A^{-1}\ C\ , \quad \text{and} \quad A = B^{-1}\ C$$

Inverse of a Matrix. The inverse of a matrix A is written as A^{-1}. The product of a matrix and its inverse is the unit matrix I,

$$A\ A^{-1} = A^{-1}\ A = I$$

Elimination Method of Finding the Inverse.[6] For a matrix with a number of zeros present, the elimination method may be simpler than the cofactor method. In general, it should not be used if there are two or more identical columns present in the original matrix.

Example:[6]

$A = [a_{ij}]$ $A^{(k)}$ denotes the k^{th} column of matrix A written in terms of $I^{(k)}$ and the coefficients of $A^{(k)}$.

$I^{(k)}$ denotes the k^{th} column of the inverse matrix A^{-1}.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix}$$

$$A^{(1)} = I^{(1)} + 2I^{(2)} + 3I^{(3)}$$
$$A^{(2)} = I^{(2)} + 4I^{(3)}$$
$$A^{(3)} = I^{(3)}$$

Solving for the I's in terms of the A's,

$$I^{(1)} = A^{(1)} - 2A^{(2)} + 5A^{(3)}$$
$$I^{(2)} = A^{(2)} - 4A^{(3)}$$
$$I^{(3)} = A^{(3)}$$

Therefore, $A^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 5 & -4 & 1 \end{bmatrix}$

Vectors and Points in Space

The method of linear programming consists of examining various points in space. A point in space may be considered by its coordinates alone or by a position vector terminating at that point, the position vector being determined by the coordinates and the origin of the system. In a 3 coordinate system (x, y, z) the unit vectors along each axis have a magnitude of unity. These vectors may be referred to as i, j, and k, and a point in space may be expressed in terms of these vectors.

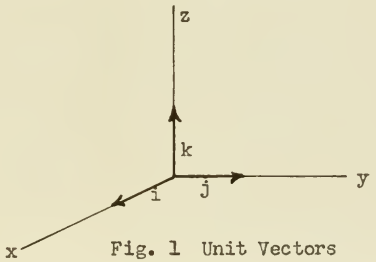


Fig. 1 Unit Vectors

Disregarding unit vectors for the moment, the segment joining two points in space in terms of other vectors is shown in Figure 2.

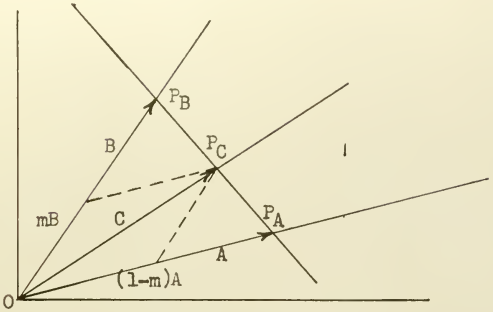


Fig. 2 Line Segment

By analytic geometry, the line segment P_B to P_A is described and generated in terms of vectors A and B by $mB + (1-m)A$, where $0 \leqslant m \leqslant 1$.

Points in Space.

A point in space may be expressed in terms of the unit vectors along each axis and a scalar.

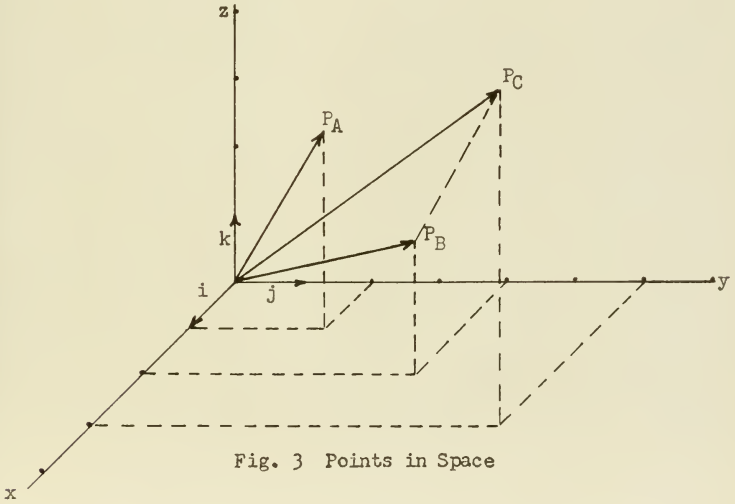


Fig. 3 Points in Space

Let $P_A = f(x,y,z)$ $\quad x = 1,\ y = 2,\ z = 3$

Then $P_A = x_A i + y_A j + z_A k = i + 2j + 3k$

Let $P_B = f(x,y,z)$ $\quad x = 2,\ y = 4,\ z = 2$

Then $P_B = x_B i + y_B j + z_B k = 2i + 4j + 2k$

$P_A + P_B = (x_A + x_B)i + (y_A + y_B)j + (z_A + z_B)k$

$P_C = P_A + P_B = (1 + 2)i + (2 + 4)j + (3 + 2)k = 3i + 6j + 5k$

The effect of multiplying each vector by a scalar is a multiplication of each unit vector.

Thus: $3P_A = 3(x_A i + y_A j + z_A k) = 3x_A i + 3y_A j + 3z_A k = 3i + 6j + 9k$

$2P_B = 2(x_B i + y_B j + z_B k) = 2x_B i + 2y_B i + 2z_B k = 4i + 8j + 4k$

$3P_A + 2P_B = (3x_A + 2x_B)i + (3y_A + 2y_B)j + (3z_A + 2z_B)k =$

$= (3 + 4)i + (6 + 8)j + (9 + 4)k = 7i + 14j + 13k$

Vectors or points P_A, P_B and P_C may be expressed as column or row vectors:

$$P_A = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad \text{or} \quad [x_A,\ y_A,\ z_A,] = [1\ 2\ 3]$$

$$P_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} \quad \text{or} \quad [x_B,\ y_B,\ z_B,] = [2\ 4\ 2]$$

$$P_C = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 5 \end{bmatrix} \quad \text{or} \quad [x_C,\ y_C,\ z_C,] = [3\ 6\ 5]$$

Because of the large number of variables involved in a linear programming problem and the use of (1, 2, 3 ...m) dimensions, the axes are expressed by numerical subscripts. For a three dimension system the axes are normally expressed as x_1, x_2 and x_3, for an m dimension

system (x_1, x_2, x_3, ...x_m). The concept of unit vectors is implicit and they are not normally written.

In the previous example the unit vectors may be expressed as row or column vectors in the form $\begin{bmatrix} i \\ j \\ k \end{bmatrix}$ or $[i\ j\ k]$

$$i = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad j = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad k = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P_A = i + 2j + 3k = 1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

By matrix multiplication and addition:

$$P_A = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Similarly, $$P_B = 2 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 4 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix}$$

$$P_C = 3 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 6 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 5 \end{bmatrix}$$

Also,

$$P_C = P_A + P_B = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 5 \end{bmatrix}$$

Charnes[1] denotes two spaces used in linear programming, a "requirement space" containing the required condition vectors and the

"solution space" containing the solution vectors. By means of linear transformation, it is possible to pass from one space to the other provided the transformation of unit vectors is known or equivalently, the transformation matrix. If $x_1, x_2, \ldots x_n$ are the original variables, and $x'_1, x'_2, \ldots x'_n$ are the new ones, they may be expressed:

$$x'_1 = a_{11}x_1 + a_{12}x_2 \cdots a_{1n}x_n$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$x'_n = a_{n1}x_1 + a_{n2}x_2 \cdots a_{nn}x_n$$

and the transformation matrix is:

$$\begin{bmatrix} a_{11}a_{12} \cdots a_{1n} \\ \cdots\cdots\cdots\cdots \\ a_{n1}a_{n2} \cdots a_{nn} \end{bmatrix}$$

Denoting the transformation matrix by A, it may be written:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ \cdot \\ \cdot \\ x'_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

and solved for in terms of either set of variables,

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = A^{-1} \begin{bmatrix} x'_1 \\ x'_2 \\ \cdot \\ \cdot \\ x'_n \end{bmatrix}$$

Letting A^{-1} be called a new matrix B with elements b_{ij} :

$$x_1 = b_{11}x'_1 + b_{12}x'_2 \cdots + b_{1n}x'_n$$
$$x_2 = b_{21}x'_1 + b_{22}x'_2 \cdots + b_{2n}x'_n$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$x_n = b_{n1}x'_1 + b_{n2}x'_2 \cdots + b_{nn}x'_n$$

Convex Sets

Theorems and proofs regarding convex sets, convex cones, and convex polyhedral cones are given by Charnes,[1] Blackwell and Girshick,[9] Gale,[10] and Gerstanhaber.[11] Gale states that the theory of polyhedral cones is equivalent to the theory of inequalities with the former being a geometric interpretation of the latter.

A convex set of points is such that if two points are in the set then the line segment joining them is also in the set. Examples of convex sets are points, lines, spheres, hyperplanes, and half spaces.[9] Hyperplanes are described by equation (2), and half spaces by equation (3).

> Theorem:[1] A linear transformation L from an n- dimensional space W takes a convex polyhedron K into a convex polyhedron L(K) , the image of K.

A complete treatment of this transformation is given in reference (1); the following figure is taken from that publication.

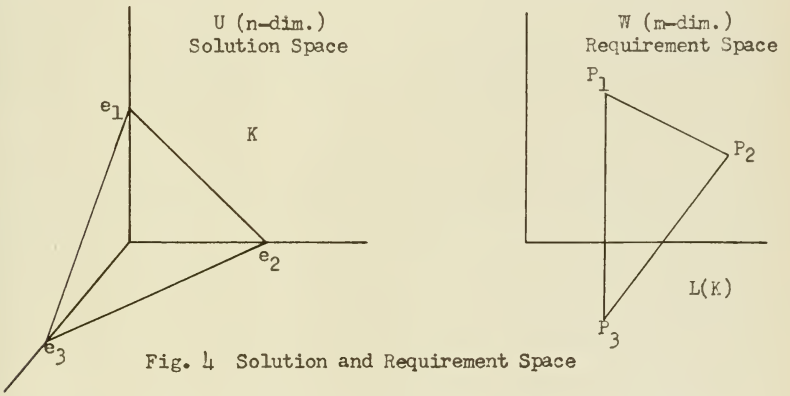


Fig. 4 Solution and Requirement Space

The configuration of L(K) is determined by the transformation matrix but will always be an image of K . Here the convex set K of 3 dimensional space is transformed into L(K) in 2 dimensional space,

the points e_1, e_2, and e_3 being transformed into points P_1, P_2, and P_3.

A convex polyhedral cone is generated by the half lines out of the origin, or rays, and the convex set. Fig. 3 contains a convex polyhedral cone with vertex at the origin and the triangular convex set formed by points P_A, P_B, and P_C. Koopmans[12] gives two equivalent definitions of convex polyhedral cones:

> Such a cone [convex polyhedral] can be defined either as the convex hull of a finite number of half-lines out of the origin (sum definition) or as the intersection of a finite number of halfspaces whose bounding hyperplanes pass through the origin (intersection definition).

An extreme point of a convex set is a point which does not lie in a segment joining two other points; they are the corners of a set such as (e_1, e_2, e_3) or (P_1, P_2, P_3) of Fig. 4. The extreme points are the points at which the linear functional, the objective, takes on its maximum (or minimum) value. The general form of linear programming, the simplex method, provides a systematic method of examining every extreme point and of obtaining the solution required by the objective.

Basis

A vector space of m dimensions is spanned or generated by a set of linearly independent vectors. The number of linearly independent points or vectors for an m dimensional space is at most m and this set of points or vectors constitutes a basis of that space. A basis must consist of a set of vectors such that every vector in the space can be written as a unique linear combination of the set. Thus in Fig. 2, the space generated by the rectangular coordinate unit vectors has a basis of its unit vectors. Also, vectors A and B can be considered

as a basis for the grid area bounded by vectors A and B , as all points on that grid space can be uniquely described in terms of the two vectors and scalars.

The basis provides an initial starting point in the solution of linear programming by the simplex method. The number of linearly independent vectors, the rank of the matrix, determines the number of dimensions involved in the requirement space. In practice the number of dimensions of the basis in the "simplex method" is determined by the number of restrictions that must be met. For example, in equation (2), this would be determined by m . In the case of the transportation method the number of variables in the basis is $m + n - 1$.

Slack Vectors

The general form for a linear program problem was given with the objective and restrictions expressed in terms of inequalities. The system of inequalities is changed to a system of equalities by adding new non-negative slack variables, sometimes called giveaway factors or disposal activities. In matrix form, the slack variables form slack vectors, and yield an identity or unit matrix which provides an initial basis. Similarly, if the objective and restraints are set up in the form of equalities, artificial vectors may be used to provide a basis. In the case of a maximizing objective, a large negative "profit" value may be associated with the artificial vectors; the value itself need not be specified.

GENERAL FORM OF LINEAR PROGRAMMING

Simplex Method

The general form of solution for any linear programming problem as developed by Dantzig,[2] is the simplex method. Charnes and Lemke[13] developed the modified method, a procedure better adapted to machine computation. Special cases of the general method have been developed such as the transportation method and the assignment method.

The general form as stated was:

(1) Maximize $\sum_{j=1}^{n} c_j x_j$

(2) Subject to $\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i \qquad i = 1, 2, 3 \ldots m$

(3) $x_j \geqslant 0 \qquad j = 1, 2, 3 \ldots n$

For $m = 3$ and $n = 3$, equations (1) and (2) take the following form:

(4) Maximize $c_1x_1 + c_2x_2 + c_3x_3 = z$

(5) Subject to $a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \leqslant b_1$

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \leqslant b_2$

$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \leqslant b_3$

(6) $x_j \geqslant 0 \qquad j = 1, 2, 3$

Adding slack variables, inequalities of (5) become:

(7) $a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 + a_{16}x_6 = b_1$

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 + a_{26}x_6 = b_2$

$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + a_{35}x_5 + a_{36}x_6 = b_3$

$x_j \geqslant 0$

Equation (7) is the form used in the simplex method of solution. The addition of slack variables also changes the linear functional (4) to:

(8) Maximize $x_1c_1 + x_2c_2 + x_3c_3 + x_4c_4 + x_5c_5 + x_6c_6 = z$

Before proceeding to a numerical example, features of (7) and (8) will be discussed; the terms used were previously described. The problem is to find the values of $x_1, x_2, x_3 \ldots x_6$ which maximize the linear functional z. There exists an infinite number of solutions, and through use of the simplex method the extreme points of the convex set are examined systematically in such a manner that the value of z is obtained for a maximum in the final tableau, frame, or table.

The column coefficients $a_{11}, a_{21}, a_{31}, \ldots; a_{16}, a_{26}$, and a_{36} represent vectors in the requirement space W and the row values $x_1, x_2, x_3, x_4, x_5, x_6$, represent a vector in the solution space U. The column coefficients a_{ij} may be written as $P_1, P_2 \ldots P_6$ and the constraints b_1, b_2, b_3 as P_o. Thus, (7) may be written as:

(9) $$P_1x_1 + P_2x_2 + P_3x_3 + P_4x_4 + P_5x_5 + P_6x_6 = P_o$$

With $P_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ a_{3j} \end{bmatrix}$

Any set of x_j's which satisfy (9) is a feasible solution and the set maximizing the value of z, in (8), is the desired optimal solution. The values of each a_{ij} and c_j may be positive, negative or equal to zero. For an objective of maximizing, meanings which may be associated with the a_{ij}, c_j, and x_j were given previously for illustration, but they are not restricted to that interpretation. As first given, the objective was to maximize profit with z in terms of dollars, but the method can be extended to any unit of measurement, time, pounds, ton miles, etc., and involving purchasing, production, stocking, selling or any similar function.

The development and proof of the simplex method in reference (2) was made under the assumption of nondegeneracy; degeneracy is the condition of having fewer than m positive values of the unknown variable at some stage of the iteration process. This restrictive assumption was first resolved for the transportation method and later resolved for the general case by Charnes.[1, 14]

The dual problem to the general case may be written:

(10) Minimize $\sum_{i=1}^{m} w_i b_i$

(11) Subject to $\sum_{i=1}^{m} w_i a_{ij} \geq c_j \qquad j = 1, 2, 3 \ldots n$

$$w_i \geq 0 \qquad i = 1, 2, 3 \ldots m$$

Here, the a_{ij}, b_i and c_j have the same meaning as in the original problem and the w_i would be in dollars per unit time, a production cost. Note the change in direction of inequalities and the change of position for the b_i and c_j. Here, slack vectors with a high penalty cost would be introduced in lieu of the original procedure. The general method of solution is the same.

Example of the Simplex Method

A problem that well illustrates the versatility of the simplex method is the production problem of references (15) and (16). The problem consists of producing two products R and S by a two stage process, initial operations must be performed in Machine Center I, and final operations may be performed in Machine Center IIA, IIAA or IIB. Machine Center IIA and IIB differ in that different unit production rates and costs are involved. In addition an amount of overtime is

available in IIA and is handled separately as IIAA. Unit production time, unit profit, and machine hours available per machine center are given in Table 1.

Table 1. Production Data

Operations	Machine Center	Product R			Product S			Maximum No. Hours Avail.
		R_1	R_2	R_3	S_4	S_5	S_6	
Initial	I	.01	.01	.01	.03	.03	.03	850
Final	IIA	.02			.05			700
	IIAA		.02			.05		100
	IIB			.03			.08	900
Unit Profit		.40	.28	.32	.72	.64	.60	

There are 3 ways of producing R, and 3 ways of producing S; therefore, each way is considered to be a different product for computation purposes. Consider the initial problem to be: maximize the total profit utilizing the available data. There are no minimum quantities of either product at this time; a requirement of this type would be handled as an additional constraint. Let x_1, x_2, x_3, x_4, x_5, and x_6 be the unknown quantities to produce of products R_1, R_2, R_3, S_4, S_5, and S_6 respectively. The problem may be stated:

Find $x_1, x_2, x_3, x_4, x_5, x_6,$

(12) Maximize $\sum_{j=1}^{6} c_j x_j = z$

(13) Subject to $\sum a_{ij} x_j \leq b \qquad i = 1, 2, 3$

$x_j \geq 0 \qquad j = 1, 2, 3, 4, 5, 6$

The a_{ij}'s are the unit production times of Table 1, the b_i's are the available machine hours, and the c_j's are the unit profit for each

product. Algebraically the problem is:

Maximize $.40x_1 + .28x_2 + .32x_3 + .72x_4 + .64x_5 + .60x_6 = z$

Subject to

(15) $.01x_1 + .01x_2 + .01x_3 + .03x_4 + .03x_5 + .03x_6 \leq 850$

(16) $.02x_1 + .05x_4 \leq 700$

(17) $.02x_2 + .05x_5 \leq 100$

(18) $.03x_3 + .08x_6 \leq 900$

The last four inequalities are changed to equalities by adding slack variables, x_7 to (15), x_8 to (16), x_9 to (17), and x_{10} to (18). The slack variables may be considered as fictitious products with corresponding c_j's , unit profit, of zero dollars and therefore will be forced out of the desired optimal solution to (14). Equations (15) through (18) may be rewritten as:

(19) $P_1x_1 + P_2x_2 + P_3x_3 + P_4x_4 + P_5x_5 + P_6x_6 + P_7x_7 + P_8x_8 + P_9x_9 + P_{10}x_{10} = P_o$

$x_j \geq 0 \qquad j = 1, 2 \ldots .10$

or $\sum_{j=1}^{10} P_jx_j = P_o$ The P_j's are column coefficients of the a_{ij} matrix of (13) and of the slack variables.

$x_j \geq 0$

The data may now be put in the form of the simplex tableau, Table 2. P_o has been placed on the left hand side of equation (19) for computational purposes. Equations (15) to (18), or equivalently (19), are in the form of (7), where some of the coefficients a_{ij} are equal to zero. The unit matrix, formed by P_7, P_8, P_9, and P_{10}, is the unit basis and provides a starting point in the simplex method. Here the requirement space is four-dimensional, and the solution space is ten-dimensional. Henceforth, zeros in a tableau are indicated by a blank space.

Table 3 is formed from the preceding tableau, Table 2, by repeating it and listing the basis vectors in the Basis column, with the corresponding c_j for each vector. The quantities of each product to produce are listed in the P_o column. Thus, the basis automatically provides an initial solution; make 850, 700, 100, and 900 units of the fictitious products x_7, x_8, x_9, and x_{10} and none of x_1, x_2, x_3, and x_4, resulting in a net profit of zero. In effect, one extreme point of the convex set has been examined; the next step is to examine the other extreme points until the objective has been obtained.

Two additional row values are inserted in Table 3: z_j and z_j-c_j. Extreme points are examined by inserting various vectors into the basis and testing each tableau, by means of the z_j-c_j value, for an optimal solution. The value of each c_j is given and its corresponding z_j is computed from $z_j = \sum_i c_i a_{ij}$, where c_i is the coefficient of the P's in the Basis column.

The following conditions for z_j-c_j and the algorithm used in the simplex method are stated without proof; derivations are contained in references (1) and (2).

Conditions of z_j-c_j :

I If any $z_j-c_j < 0$, and for every i, $x_{ij} \leq 0$ then z is infinite.

II If any $z_j-c_j < 0$, and for some i, $x_{ij} > 0$, the objective has not been obtained and the simplex procedure is continued.

III If all $z_j-c_j \geq 0$, then the value of z_o is a maximum and the objective has been obtained.

Conditions II and III are normally encountered as condition I may be excluded when setting up the problem. In continuing the simplex procedure, following condition II, each successive value of the linear

functional z increases until the objective is obtained. It should be noted that if there exists a $z_j - c_j = 0$, for some P_j not in the basis of the final answer, there are additional combinations of real products available but whose objective value will equal the present z_o. For all P's in the basis, $z - c = 0$ is always true.

The next step is to bring into the basis some new column P_k vector. Any P_j having a negative $z-c$ is admissible, but the procedure is to bring in that P having the largest negative $z-c$ value such as P_4. To determine which row vector, P_r, to remove from the basis, divide each element in the P_o column by each corresponding positive a_{ij} appearing in the P_k column coming into the basis. The minimum positive ratio so formed determines the row vector to be removed. This ratio must be positive since the original conditions required the components of the P_o vector to be non-negative, equations (19).

For illustration the following general tableau is given:

Vector coming in

Basis	P_o	P_j		P_k
P_i	X_i	a_{ij}		a_{ik}
P_r	X_r	a_{rj}		a_{rk}
P_i	a'_{io}	a'_{ij}		a'_{ik}
P_k	$\frac{a_{xr}}{a_{rk}}$	$\frac{a_{rj}}{a_{rk}}$		$\frac{a_{rk}}{a_{rk}} = 1$

Vector being removed ←

Old Tableau

Vector coming in →

New Tableau

Let P_j = any column vector.

P_k = column vector coming into the basis.

P_r = row vector being removed from the basis.

a_{rk} = coefficient determined by the intersection of the P_k column and P_r row.

In setting up the successive tableaus, the vector coming into the basis is determined first. P_k and P_r are determined as previously stated and a_{rk} follows immediately. The P_k row may be written directly by dividing through the old entire P_r row by a_{rk}.

All other rows may be determined by the following formula or algorithm. Derivations are contained in references (1) and (17).

Let a'_{ij} = any other new element.

a_{ij} = corresponding old element.

a_{rj} = old element in the same column and in the row removed.

a_{ik} = old element in the same row and in the column entering.

a_{rk} = old element in the row removed and in the column entering.

$$a'_{ij} = a_{ij} - a_{rj}\frac{a_{ik}}{a_{rk}} \qquad (i \neq k)$$

The new tableau is completed using the above algorithm, z_j's and the corresponding $z_j - c_j$ are computed and inserted in the tableau. Also the new profit value is given by $z'_0 = z_0 - \theta(z_k - c_k)$ where:

$$\theta = \min_{\substack{i \\ a_{ik} > 0}} \frac{X_i}{a_{ik}},$$

the minimum positive ratio previously described in determining P_r.

The maximum negative value of $z_j - c_j$ determines the next P_k; the process is repeated until such time as there are no negative values of $z_j - c_j$, and the final tableau contains the optimum solution. Also the new $z_j - c_j$, $z'_j - c_j$, may be obtained from:

$$z'_j - c_j = z_j - c_j - \frac{a_{rj}}{a_{rk}}(z_k - c_k)$$

In summary the $z_j - c_j$ are needed to determine the P_k to come into the basis. By choosing the most negative value of $z_j - c_j$, it often results in a quicker solution, but not always. Knowing P_k, it is then

necessary to determine the P_r to be removed from the basis. These two phases are the key to almost all special methods of linear programming.

Continuing the example, the next tableau, Table 4, is constructed. The most negative $z_j - c_j$ is column P_4, the P_k indicated by the vertical arrow. The P_r, row to be removed, is determined as follows:

$$\min_i \frac{X_i}{a_{ik}}, \quad a_{ik} > 0$$

$$\frac{850}{.03} = 28{,}333 \text{ for } P_7, \quad \frac{700}{.05} = 14{,}000 \text{ for } P_8$$

Therefore, P_8, the minimum ratio row, becomes the P_r. Column headings $P_o \ldots P_{10}$ and the c_j values in the corresponding columns remain the same and are omitted in succeeding tables for brevity.

Each element in the old P_8 row is divided through by the a_{rk} value of .05, yielding 14,000 in the P_o column, .4 in the P_1 column, 1 in the P_4 column, and 20 in the P_8 column.

The elements in all remaining rows are computed by the algorithm,

$$a'_{ij} = a_{ij} - \frac{a_{rj}}{a_{rk}} a_{ik} :$$

$$a'_{70} = a_{70} - \frac{a_{80}}{a_{84}} a_{74}$$

$$a'_{70} = 850 - \frac{(700)}{(.05)} .03 = 430$$

$$a'_{71} = a_{71} - \frac{a_{81}}{a_{84}} a_{74}$$

$$a'_{71} = .01 - \frac{(.02)}{(.05)} .03 = -.002$$

Each element in the new table is determined as above. It should be noted that use of the algorithm sets up a square pattern with the initial a_{rk} determined. The pattern is indicated as follows:

Essentially the algorithm states: The new element equals the old element minus the product of the diagonals divided by the a_{rk} . Element a_{78} in Table 4 is obtained from Table 3 by the above pattern:

$$a_{78} = 0 - \frac{(.03)(1)}{.05} = -.6$$

The diagonal product may be left to right, right to left, and up or down depending upon the relative position of the a_{rk} . The resulting tableau is given in Table 4. A shortcut to avoid computing each element when zeros are present will be discussed later.

The z_j's are computed for each column, and the corresponding c_j's are subtracted yielding the z_j-c_j row. The z_j-c_j are scanned for the most negative value, and the process is repeated until Table 9 has been obtained, the optimum solution. Thus, the solution calls for the number of products R_1, R_2, R_3 as indicated in the P_o column with the corresponding profit in the z_j-c_j entry. It also calls for 150 units of product X_7 , but this was a fictitious product and is ruled out. Product S is excluded since it does not appear in

Table 2

↓c →			.40	.28	.32	.72	.64	.60				
	Basis	P_o	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
		850	.01	.01	.01	.03	.03	.03	1	0	0	0
		700	.02	0	0	.05	0	0	0	1	0	0
		100	0	.02	0	0	.05	0	0	0	1	0
		900	0	0	.03	0	0	.08	0	0	0	1

Table 3

↓c →			.	.40	.28	.32	.72	.64	.60				
	Basis	P_o		P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
0	P_7	850		.01	.01	.01	.03	.03	.03	1			
0	P_8	700		.02			.05				1		
0	P_9	100			.02			.05				1	
0	P_{10}	900				.03			.08				1
	z_j												
	$z_j - c_j$			-.40	-.28	-.32	-.72	-.64	-.60				

(P_r ←)

Table 4

	P_7	430	-.002	.01	.01		.03	.03	1	-.6		
.72	P_4	14,000	.4			1				20		
	P_9	100		.02			.05				1	
	P_{10}	900			.03			.08				1
	z_j	10,080	.288			.72				14.4		
	$z_j - c_j$	10,080	-.112	-.28	-.32		-.64	-.60		14.4		

(P_k →, P_r ←)

Table 5

	P_7	370	-.002	-.002	.01			.03	1	-.6	-.6	
.72	P_4	14,000	.4			1				20		
.64	P_5	2,000		.4			1				20	
	P_{10}	900			.03			.08				1
	z_j	11,360	.288	.256		.72	.64			14.4	12.8	
	$z_j - c_j$	11,360	-.112	-.024	-.32			-.60		14.4	12.8	

(P_k →, P_r ←)

Table 6

	C ↓ →			.40	.28	.32	.72	.64	.60				
		Basis	P_o	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
		P_7	32.5	-.002	-.002	-1/800				1	-.6	-.6	-.375
P_r ←	.72	P_4	14,000	.4			1				20		
	.64	P_5	2,000		.4			1				20	
P_k →	.60	P_6	11,250			.375			1				12.5
		z_j	18,110	.288	.256		.72	.64	.60		14.4	12.8	7.5
		z_j-c_j	18,110	-.112	-.024	-.095					14.4	12.8	7.5

Table 7

		P_7	102.5		-.002	-1/800	.005			1	-.5	-.6	-.375
P_k →	.40	P_1	35,000	1			2.5				50		
	.64	P_5	2,000		.4			1				20	
P_r ←	.60	P_6	11,250			.375			1				12.5
		z_j	22,030	.4	.256	.225	1	.64	.60		20	12.8	7.5
		z_j-c_j	22,030		-.024	-.095	.28				20	12.8	7.5

Table 8

		P_7	140		-.002		.005		1/300	1	-.5	-.6	-1/3
	.40	P_1	35,000	1			2.5				50		
P_r ←	.64	P_5	2,000		.4			1				20	
P_k →	.32	P_3	30,000			1			8/3				100/3
		z_j	24,880	.4	.256	.32	1	.64	.853		20	12.8	10.6
		z_j-c_j	24,880		-.024		.28		.253		20	12.8	10.6

Table 9 - Final Solution

	Basis	P_o	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
	P_7	150				1/200	1/200	1/300	1	-.5	-.5	-.33
.40	P_1	35,000	1			2.5				50		
.28	P_2	5,000		1			2.5				50	
.32	P_3	30,000			1			8/3				100/3
	z_j	25,000	.40	.28	.32	1	.70	.853		20	14	10.6
	z_j-c_j	25,000				.28	.06	.253		20	14	10.6

the basis. The program is summarized in Table 10.

Table 10

Optimal Program - Number of Units

Product R	Product S
R_1 (Centers I, IIA). . 35,000 units	S_4 N
R_2 (I, IIAA) . . . . . 5,000 units	S_5 o
R_3 (I, IIB). . . . . 30,000 units	S_6 n
	e
Total profit $25,000	

Table 11 summarizes the program in number of hours using the data of Tables 1 and 10.

Table 11

Optimal Program - Number of Hours

Operation	Machine Center	Product R			Product S			Hours Used	Hours Avail.	Surplus Hours
		R_1	R_2	R_3	S_4	S_5	S_6			
Initial	I	350	50	300	0	0	0	700	850	150
Final	IIA	700			0			700	700	0
	IIAA		100			0		100	100	0
	IIB			900			0	900	900	0

It should be noted that when a zero appears in the P_r row of any tableau, the entire column in which it appears will be identical in the following tableau. Similarly, if a zero appears in the P_k column, the old row repeats itself in entirety. Referring to Table 6, zeros appear in row P_4 for columns P_2, P_3, P_5, P_6, P_7, P_9, P_{10}, and those columns are identically repeated in Table 7. In Table 6, zeros appear

in the P_k column, P_1, for rows P_5 and P_6. Thus, rows P_5 and P_6 in Table 6 are identically repeated in Table 7. Because of the many zeros encountered in a typical linear programming solution, the algorithm need not be used for every element when the above conditions are present. The general form, the simplex method, has been criticized as a time consuming process; however, by properly utilizing various short cuts, the time involved is significantly reduced. A review of Tables 3 to 9 will quickly verify the above statement.

An inverse matrix is contained in each tableau if the initial tableau contains an identity matrix.

> Theorem:[6] If, in an initial tableau we have an identity matrix, then in this position in subsequent tableaus will be the inverse of the current basis.

Referring to Tables 3 and 4, the matrix formed by P_7, P_8, P_9, and P_{10} column of Table 4, is the inverse matrix of the new basis, P_7, P_4, P_9, P_{10} of Table 4, as given by the P_7, P_4, P_9, and P_{10} columns of Table 3.

Table 3

P_7,	P_4,	P_9,	P_{10}
1	.03	0	0
0	.05	0	0
0	0	1	0
0	0	0	1

Table 4

P_7,	P_8,	P_9,	P_{10}
1	-.6	0	0
0	20	0	0
0	0	1	0
0	0	0	1

$=$

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Similarly, by the same theorem, in each succeeding tableau an inverse matrix will be in the same position as the previous identity matrix. In Table 5 a unit matrix is given by P_7, P_4, P_5, and P_{10}.

Therefore, in the same columns of Table 6, the inverse of the new basis, P_7, P_4, P_5, P_6 of Table 6, will be given.

Table 5

P_7	P_4	P_5	P_6
1	0	0	3/100
0	1	0	0
0	0	1	0
0	0	0	8/100

Table 6

P_7	P_4	P_5	P_{10}
1	0	0	-3/8
0	1	0	0
0	0	1	0
0	0	0	25/2

$=$

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Carrying the theorem further leads to the modified simplex method developed by Charnes and Lemke.[13] This also leads to the basis for every "adjacent exteme point" method to date and most of the special methods of solution. In the modified simplex method, the algorithm need only be applied to that part of the tableau under the unit vectors, and by one equation the x_i's and the $z_j - c_j$ may be developed. The P_k is then determined, a_{ik} and $\frac{x_i}{a_{ik}}$ developed, P_r determined and the simplex algorithm applied as before.

Referring to the last theorem, it is possible to develop all elements in subsequent tableaus by its use. Using Table 4 for example, P_o may be written:

$$P_7x_7 + P_4x_4 + P_9x_9 + P_{10}x_{10} = P_o$$

or in matrix notation,

$$\begin{bmatrix} P_7 & P_4 & P_9 & P_{10} \end{bmatrix} \begin{bmatrix} x_7 \\ x_4 \\ x_9 \\ x_{10} \end{bmatrix} = P_o$$

Let $B = [P_7\ P_4\ P_9\ P_{10}]$, and $x = \begin{bmatrix} x_7 \\ x_4 \\ x_9 \\ x_{10} \end{bmatrix}$

Then, $Bx = P_o = \begin{bmatrix} 850 \\ 700 \\ 100 \\ 900 \end{bmatrix}$

Multiplying through by the inverse of B, B^{-1} , the latter equation becomes:

$$B^{-1}Bx = B^{-1}P_o$$

or $$x = B^{-1}P_o$$

Therefore, it is possible to obtain the new x_i values by means of the inverse of the current basis matrix in the new tableau from the P_o column vector in the old tableau. For example the x_i in the P_o column of Table 4 can be obtained as follows:

$$x_7 = 1(850) - .6(700) + 0(100) + 0(900) = 430$$

$$x_4 = 0(850) + 20(700) + 0(100) + 0(900) = 14{,}000$$

$$x_9 = 0(850) + 0(700) + 1(100) + 0(900) = 100$$

$$x_{10} = 0(850) + 0(700) + 0(100) + 1(900) = 900$$

Similarly new column elements can be developed using the old P_j column and the inverse of the current basis matrix. Elements of P_1 in Table 4 illustrate this:

$$a_{11} = 1(.01) - .6(.02) + 0(0) + 0(0) = -.002$$

$$a_{21} = 0(.01) + 20(.02) + 0(0) + 0(0) = .4$$

$$a_{31} = 0(.01) + 0(.02) + 1(0) + 0(0) = 0$$

$$a_{41} = 0(.01) + 0(.02) + 0(0) + 1(0) = 0$$

It is to be noted that the inversion of a large matrix becomes increasingly difficult as the size of the matrix increases.

The usefulness of linear programming in analyzing cost data for changes in initial restrictions such as minimum product requirements, available machine time, unit production rates, and unit profits is well illustrated by this problem.

Initially, there were no minimum requirements of product S to meet obligations such as market demands. Assume a minimum of 8,000 units of product S are required in addition to the initial conditions. The problem need not be reworked as the necessary data is contained in Table 9, the former solution.

The $z_j - c_j$ row represents "opportunity costs." Thus the process given by P_5 of Table 9 has the least value, .06, a value that will affect the total profit the least for any change to the production program. The next succeeding values for product S are .253 and .28, for P_6 and P_4 respectively. Thus, P_5 represents the best choice and will necessitate reducing the amount of x_2 produced, represented by P_2 in the basis. However, the initial condition on x_i was that $x_i \geqslant 0$.

Therefore, $2.5S_5 \leqslant 5000$, and for maximum benefit the inequality sign is dropped yielding: $2.5S_5 = 5000$, or $S_5 = 2000$.

Since 8,000 units of S are required, the remaining 6,000 units must be S_4 , S_6 or some combination thereof. The next lowest value of the $z_j - c_j$ row is .253 of P_6 , requiring a reduction of x_3 produced, represented by P_3 in the basis.

Therefore, $\frac{8}{3}S_6 \leqslant 30{,}000$, and this condition is satisfied for $S_6 = 6000$. In addition, x_3 is reduced by: $(8/3)(6000) = 16{,}000$.

Table 12 summarizes the above results. In the first step the profit was reduced by (2000)($.06) = $120, and in the second step by (6000)($.253) = $1520.

Table 12

Optimal Program - Number of Units

Product R	Product S
R_1 (I, IIA) . . . 35,000	S_4 0
R_2 (I, IIAA). . . 0	S_5 2000
R_3 (I, IIB) . . . 14,000	S_6 6000

Total Profit = 25,000 - (120 + 1520) = $23,360

The z_j-c_j row of Table 9 may also be utilized to analyze the effect of changes to the hours available in each machine center. For z_j-c_j , j = 7, 8, 9, 10, initially introduced as a result of the 4 slack variables x_7, x_8, x_9, and x_{10} , the "opportunity costs" represent values of additional profit for each additional machine center unit time of centers I, IIA, IIAA, and IIB respectively.

From Table 9 it is more profitable to make additional time available in centers IIA, IIAA, and IIB as indicated by the z_j-c_j of 20, 14, and 10.6. 150 hours of surplus time is already available in center I and verified by $z_7-c_7 = 0$, the least profitable center to allocate additional time. Assuming the 150 surplus hours available are to be utilized but not exceeded, the upper limits for IIA, IIAA, and IIB are obtained as follows:

Center IIA $\quad 150 + H_8 (-.5) \geq 0$, therefore $H_8 \leq 300$ hrs.

Center IIAA $\quad 150 + H_9 (-.5) \geq 0$, therefore $H_9 \leq 300$ hrs.

Center IIB $\quad 150 + H_{10}(-.33) \geq 0$, therefore $H_{10} \leq 450$ hrs.

Allocation of additional hours to Center IIA, IIAA, and IIB, within the above limits, will increase the original total profit as follows:

New profit = $25{,}000 + 20H_1 + 14H_2 + 10.5H_3$

for $0 \leq H_8 \leq 300$, $0 \leq H_9 \leq 300$, $0 \leq H_{10} \leq 450$

Change to unit production rates of Table 1, as a result of new tools or machines, may require a new solution of the entire problem. However, the use of the final solution of Table 9 in analyzing such a change can be shown in this problem. Assume that the machining time per unit in Center IIA and IIAA for product R can be reduced from .02 to .0175. The most profitable methods are R_1, R_3, and R_2 respectively. Allocating the total time available in Center IIA, 700 hours, requires 400 hours in Center I. The next most profitable method is for product R_3, and 900 hours of Center IIB are allocated. This requires 300 hours in Center I for product R_3. The 100 hours of Center IIAA is allocated for product R_2 which in turn requires 57.1 hours in Center I. Table 13 summarizes the new program. Profits are computed from the number of parts used in determining Center I hours and the initial data. The total profit does not take into account the improvement cost but does provide a figure of $2,198.80 as a criterion for judging the worth of such an improvement.

Table 13

Optimal Program for Machine Time of $IIA_R = .0175$

Operation	Machine Center	Product R			Product S			Hours Used	Hours Avail.	Surplus Hours
		R_1	R_2	R_3	S_4	S_5	S_6			
Initial	I	400	57.1	300	0	0	0	757.1	850	92.9
Final	IIA	700			0			700	700	0
	IIAA		100			0		100	100	0
	IIB			900			0	900	900	0

Total Profit: $16{,}000 + 1598.80 + 9{,}600 = \$27{,}198.80$

Changes in unit profits may or may not require a new solution of the problem. In either event Table 9 may be utilized as an initial starting point by using the elements of that table and the new unit profits. The new z_j and z_j-c_j are computed; if all z_j-c_j are positive the program is still optimum. If any z_j-c_j is negative, the simplex procedure is carried out until all such values are positive.

Assume that after obtaining the program of Table 9, it is necessary to reduce the sales price of product R by \$.04 to meet new competitive demands. Table 14 is obtained from Table 9 by repeating the elements and inserting new c_j's , \$.04 lower, for R_1, R_2, and R_3 . Under the old program of Tables 9 and 14, and the new unit profit figures, the profit is \$22,200, a reduction of \$2,800. With the new program of Table 15, the profit is \$22,280, a reduction of \$2,720 from that of the original solution.

Table 15 is obtained from Table 14 in three steps. The P_k , P_5 , is determined by the z-c value of -.04, and P_r from the minimum ratio of $\frac{x_i}{a_{i5}}$. In Table 15, P_5 row is calculated by dividing the old P_2 row by the a_{rk} of 2.5. New rows P_1 and P_3 are identical as in Table 14 in view of the zeros present in the P_r and P_k , rows and columns rule. Thus, the "square pattern" of the algorithm need only be applied to new row P_7 . z_j and z_j-c_j are then computed, all values of the latter are positive and therefore an optimal program exists.

Table 14

↓c →			.36	.24	.28	.72	.64	.60				
	Basis	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
	P_7	150				1/200	1/200	1/300	1	-.5	-.5	-.33
.36	P_1	35,000	1			2.5				50		
← .24	P_2	5,000		1			(2.5)				50	
.28	P_3	30,000			1			8/3				100/3
	z_j		.36	.24	.28	.90	.60	.746		18	12	9.3
	z_j-c_j					.28	-.04	.146		18	12	9.3

Table 15

	P_7	140		-1/500		1/200		1/300	1	-.5	-.6	-.33
.36	P_1	35,000	1			2.5				50		
→ .64	P_5	2,000		.4			1				20	
.28	P_3	30,000			1			8/3				100/3
	z_j		.36	.256	.28	.9	.64	.746		18	12.8	9 1/3
	z_j-c_j			.016		.18		.146		18	12.8	9 1/3

The usefulness of linear programming in cost analysis is an important characteristic that should not be overlooked. The manufacturing process involved, depending upon its complexity, may cause difficulty in normal cost analysis. In an oil refinery problem, it was necessary to produce a minimum quantity of a product for goodwill purposes, and at little or no profit. Because of the process involved, it was found that to produce only the minimum quantity, the resulting costs were higher than they were to produce a quantity above the minimum.

Reference (17) illustrates a problem in planning an experimental program, solved by linear programming and game theory. Alternatives such as installing the plant without preliminary experimental tests, with bench scale, and pilot plant tests were considered.

TRANSPORTATION METHOD[6, 13, 17, 19]

The transportation method is a special case of the general simplex method. It was originally considered by Hitchcock and by Koopmans. Assume there are $a_1, a_2 \ldots a_m$ homogeneous products at origins $O_1, O_2 \ldots O_i \ldots O_m$. These quantities are to be completely distributed to destinations $D_1, D_2 \ldots D_j \ldots D_n$ in quantities $b_1, b_2, \ldots b_j \ldots b_n$. Total distribution requires that $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$. Let the unit cost of transportation from O_i to D_j be c_{ij} and the amount to be transported be x_{ij}. The objective is to minimize the total cost of transportation. A special case of the above method is known as the "assignment method" where all of the a_i's and b_j's are equal to unity. The transportation problem takes the following form:

(1) Minimize $\sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij} x_{ij}$

(2) Subject to $\sum_{j=1}^{n} x_{ij} = a_i \qquad i = 1, 2, 3 \ldots m$

(3) $\sum_{i=1}^{m} x_{ij} = b_j \qquad j = 1, 2, 3 \ldots n$

$x_{ij} \geq 0$

If a problem can be put in the above form, it is possible to use the transportation method. Written out in general form for three origins and four destinations, equations (2) and (3) take the following form:

$$
\begin{aligned}
x_{11} + x_{12} + x_{13} + x_{14} &= a_1 \\
x_{21} + x_{22} + x_{23} + x_{24} &= a_2 \\
x_{31} + x_{32} + x_{33} + x_{34} &= a_3 \\
x_{11} + x_{21} + x_{31} &= b_1 \\
x_{12} + x_{22} + x_{32} &= b_2 \\
x_{13} + x_{23} + x_{33} &= b_3 \\
x_{14} + x_{24} + x_{34} &= b_4
\end{aligned}
$$

This could also represent three products and four machine operations. The transportation problem always has the above symmetrical form, equal number of rows and diagonals. In addition, there are exactly two coefficients of unity in each column and all other entries are zero.

In a production problem for example, the coefficients would rarely be unity for production rates. However, it is possible to change the coefficients to unity and obtain an approximate solution by two basic operations:[6]

1. Multiplication of an inequation by a positive constant.
2. A change in scale for certain of the variables.

This procedure results in an approximate solution due to the rounding errors introduced. "Normalizing" is the term used to describe the above operation which results in working with equivalent production units in place of the original production units. A complete description for handling the "normalizing" procedure is contained in reference (19). It also contains numerous examples as well as a general discussion of linear programming and the transportation method in particular. As in the simplex problem, artificial vectors may be used to handle problems where the total amount available does not equal the total amount required.

Fictitious origins and destinations with fictitious costs permit an optimal program; all fictitious routes are forced out in final solution. By the above process a mathematical analogy may be made, where originally there was no apparent physical analogy. The construction of the mathematical model requires complete familiarity with the different methods of linear programming.

The following problem illustrates the "stepping stone" method, originally developed by Dantzig (18) and described by Charnes and Cooper.[6, 20] It can be shown that the $m + n$ equations, (2) and (3), constitute $m + n - 1$ independent equations in $m \times n$ unknowns.[18] These equations therefore form a basis, and the minimizing solution will involve at most $m + n - 1$ routes with positive shipments.

The stepping stone method consists in setting up an initial solution by means of the basis. The a_{ij} elements are $+1$, -1, or 0. Thus in tabular form the x_i is used directly in forming the minimum positive ratio $\frac{x_i}{a_{ik}}$. The c_{ij}, cost of each ij route, is inserted in the small corner boxes of Table 16 for ease in computation. Quantities available are listed in the right hand column, and quantities required in the bottom row. Initially an arbitrary number is assigned to some row, normally the row containing the most stepping stone positions, the vectors in the basis. The K_j's for stepping stone columns are then determined from equation (4):

(4) $\quad R_i + K_j = c_{ij}$ where R_i = Row number
K_j = Column number

Row and column values can then be computed for the remaining stepping stone positions by equation (4).

Row and column values for the non-stepping stone positions, the vectors not in the basis, can be determined from R_i's and K_j's available. For the non-stepping stone positions the z_{ij} is obtained from equation (5):

(5) $$z_{ij} = R_i + K_j$$

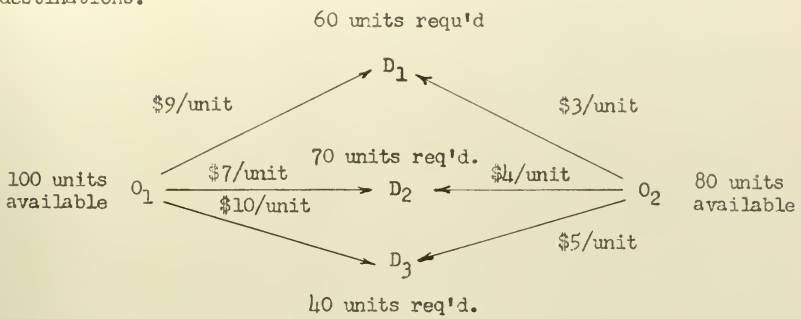
From equation (5), the $z_{ij}-c_{ij}$ is computed for each cell not in the basis. The non-stepping stone cell with the largest positive value of $z_{ij}-c_{ij}$ is then brought into the solution, and the stepping stone cell with the minimum positive $\frac{x_i}{a_{ik}}$ ratio is removed. To bring a new vector into the basis, it must be brought in by a path via the stepping stone positions by rook moves, as in chess. The amount brought in is added or subtracted alternately to the basis positions used. The amount brought in is subtracted from the positive cell and added to the negative cell. In the result of a tie for $\frac{x_i}{a_{ik}}$, remove the most northwesterly value, and if the tie occurs in the same row remove the most westerly value.

New R_i and K_j values are determined, the non-stepping stone $z_{ij}-c_{ij}$ determined, and the process repeated until all of the $z_{ij}-c_{ij}$ are negative or equal to zero. For non-stepping stone cells with $z_{ij}-c_{ij}$ equal to zero, it indicates an alternate route with the total cost equal to that of the present solution.

Example of Transportation Method

For a problem with two origins, three destinations, and corresponding transportation costs as given below, the use of fictitious values

is shown for the case of $\Sigma a_i \neq \Sigma b_j$. The objective is to obtain the lowest total transportation cost and fulfill all requirements of the destinations.



Algebraically the problem takes the following form:

$$\text{Minimize} \quad 9x_{11} + 7x_{12} + 10x_{13} + 3x_{21} + 4x_{22} + 5x_{23}$$

$$\text{Subject to} \quad x_{11} + x_{12} + x_{13} \leq 100$$

$$x_{21} + x_{22} + x_{23} \leq 80$$

$$x_{11} + x_{21} = 60$$

$$x_{12} + x_{22} = 70$$

$$x_{13} + x_{23} = 40$$

$$x_{ij} \geq 0$$

Since there are 10 more units available than are required, a fictitious destination is set up, and a zero transportation cost assigned in order to insure such a route is utilized for the excess supply.

With the fictitious destination, the above equations take the following form:

$$9x_{11} + 7x_{12} + 10x_{13} + 0x_{1F} + 3x_{21} + 4x_{22} + 5x_{23} + 0x_{2F}$$

$$x_{11} + x_{12} + x_{13} + x_{1F} = 100$$

$$x_{21} + x_{22} + x_{23} + x_{2F} = 80$$

$$x_{11} + x_{21} = 60$$

$$x_{12} + x_{22} = 70$$

$$x_{13} + x_{23} = 40$$

$$x_{1F} + x_{2F} = 10$$

$$x_{ij} \geq 0$$

Due to the characteristics of this type problem, the problem may be reduced to the tabular form of Table 16 for computation purposes.

Table 16

	D_1	D_2	D_3	D_4	
O_1	9 x_{11}	7 x_{12}	10 x_{13}	0 x_{1F}	100
O_2	3 x_{21}	4 x_{22}	5 x_{23}	0 x_{2F}	80
	60	70	40	10	180

Table 17 is constructed from the data of Table 16. Since the number of vector elements in the basis is five ($m + n - 1$), an estimated solution may be written down immediately. For a systematic approach, particularly with a large matrix, it is convenient to start at the northwest corner, work across and down, balancing requirements at each step. This procedure is used in Table 17, total cost is $1140.

Table 17

R_i \ K_j	6	4	5	0	
3	9 60 +	7 40 -	10	0	100
0	3	4 30 +	5 40	0 10	80
	60	70	40	10	180

Rules in establishing row and column values, R_i and K_j, for the stepping stone positions are summarized below:[6]

1. Arbitrarily assign 0 to a row.
2. Move eastward along the row until you encounter the first vector in the basis.
3. Assign to the column the value necessary to cause the row and column sum to equal the cell coefficient. [Equation (4)]
4. Duplicate the process for all cells in the row which are included in the basis.

To establish row and column values for cells not in the basis:[6]

5. Proceed to determine any other row number by selecting a vector within the basis for which a column number is available.
6. Carry through this procedure until all row and column numbers have been determined.

Applying the procedure to Table 17, assign zero to R_2, this determines the K_2, K_3, K_4 of 4, 5, 0 respectively by equation (4). R_1 may then be found from K_2 and equation (4), equal to 3. K_1 is then determined by the same equation. The z_{ij}'s for cells 13, 14, and 21, are found by equation (5); the corresponding c_{ij}'s are subtracted, giving values of - 2, + 3, and + 3 respectively. Cell 21 with the largest positive $z_{ij}-c_{ij}$, + 3, is the vector to be brought into the basis, and cell 22 with the lowest positive $\frac{x_i}{a_{ik}}$ ratio, 30, is to be removed from

the basis. Here, two cells resulted in a tie by the +3 value. In a matrix, rows may be interchanged and columns may be interchanged. Interchanging row 1 and 2 would result in cell 21 becoming the most northwesterly cell, applying the rule for case of a tie, requires cell 21 to be brought in. Cell 21 is brought in via the stepping stones by rook moves. Note, P_{21} may be written:

$$P_{21} = \overset{(30)}{P_{22}} - \overset{(40)}{P_{12}} + \overset{(60)}{P_{11}}$$

The a_{ik}'s are +1, -1, or 0, and the plus and minus signs within each cell refer to each a_{ik} of the path. To determine θ, only the P's with the +1 coefficients are considered, P_{22} and P_{11}. Minimum θ, $\frac{x_i}{a_{ik}}$, is $\frac{30}{1}$; the amount to be removed is 30 for a reduction in cost of \$3 X 30 or \$90. Total cost is \$1140 - \$90 = \$1050. The new program is given in Table 18.

Table 18

R_i \ K_j	3	1	5	0	
6	9 30 +	7 70	10	0	100
0	3 30 -	4	5 40	0 10 +	80
	60	70	40	10	180

New row and column values are determined and shown in Table 18. The $z_{ij}-c_{ij}$ for non-stepping stone positions show that Cell 14, a difference of +6, must be brought in through stepping stones P_{11}, P_{21}, and P_{24} with minimum θ at P_{24}; therefore P_{24} is removed. Reduction in cost is \$6 X 10 = \$60, total cost of \$1050 - \$60 or \$990. The

program is given in Table 19.

Table 19

R_i \ K_j	9	7	11	0	
0	[9] 20+	[7] 70	[10]	[0] 10	100
- 6	[3] 40 -	[4]	[5] 40+	[0]	80
	60	70	40	10	180

New row and column values are determined, given in Table 19, the $z_{ij}-c_{ij}$ differences computed, P_{13} is brought in, P_{11} removed, and the cost reduction of \$20 is obtained, \$1 X 20. The program is given in Table 20 at a total cost of \$970.

Table 20

R_i \ K_j	8	7	10	0	
0	[9]	[7] 70	[10] 20	[0] 10	100
- 5	[3] 60	[4]	[5] 20	[0]	80
	60	70	40	10	180

New row and column values are assigned and given in Table 20. The $z_{ij}-c_{ij}$ for all non-stepping stone positions are negative and an optimal program has been obtained. If any of the $z_{ij}-c_{ij}$ for the above positions were equal to zero, such a cell would represent an alternate route but at the same total cost.

Example of Transportation Method, Fictitious Route[6]

To illustrate other considerations concerning the transportation method, the following example is given. Procedural explanation is

omitted except in calling attention to specific considerations. Tables 21 and 22 show the successive stages obtained through the steps previously discussed. The problem consists of 3 origins and 5 destinations.[6] Cost data and quantities are given in Table 21. The allocation to routes was made by starting in the northwest corner and proceeding across and down. The requirements are fulfilled by six cell values, excluding the zero. However, in this method the number of vector elements in the basis for an m by n matrix is $(m + n - 1)$. Therefore, 7 vector elements must be used $(3 + 5 - 1)$. The optimal program may consist of less than $(m + n - 1)$ routes as in this example; however, during the iteration process all possible basis vectors must be considered at each stage. To fulfill the requirement of 7 vector elements in the basis, zero is arbitrarily assigned to $O_2 D_2$. The zero could have been assigned to any of the 9 remaining cells available, and this would require different row and column numbers than given in Table 21. The transportation method is carried out, P_{24} removed and P_{25} brought in. The optimal program is then given in Table 22.

There are 5 non-stepping stone cells with $z_{ij}-c_{ij}$ equal to zero, cells 13, 21, 31, 32, and 33; therefore, those cells represent alternate available routes at the same cost as the program of Table 22, $230. When there is more than one basic solution the other solutions consist of a convex linear combination of the basic solution. Because the cost data is a matrix, if a number, positive or negative, is added to any entire row or to any entire column, the new problem is equivalent to the old for an optimal program.[6] By subtracting 3, 4, 5, 6, and 2 from columns 1, 2, 3, 4, 5 respectively, adding 2 to row 1 and adding 1 to row 2, the "cost" data of Table 23 is obtained. The route can be programmed

according to the new relative "costs"; the - 5 "cost," cell 34, may be viewed as a subsidy route with the maximum amount to be shipped via that route. The + 5 "cost" of cell 15 is avoided by use of the many alternate routes of 0 "cost." The program may be set up to meet all requirements and the total cost obtained through the use of the original cost data.

The transportation method permits cost analysis similar to the simplex method, for factors such as: what cost increase is permissible for some particular route to meet competitive conditions and still maintain an optimum program; where to ship, produce, store; and similar functions. The objective of maximizing some functional may also be used in the transportation method. Symonds[17] discusses an oil refinery problem of allocating crude oils to several refineries in order to obtain maximum possible profit. The problem, a 5 X 10 matrix, is solved by 4 iterations.

In the two previous illustrations, a separate table was used in each iteration for clarity. In practice, all iterations may be carried out using one table by crossing out old row and column numbers and cell elements changed. It is a characteristic of this method that there will be no undue buildup of changed values in any one cell. Furthermore, there is no need of rechecking calculations for errors as a positive check is provided on completion of each iteration by the use of new row and column values.

Table 21[6]

	D_1 2	D_2 3	D_3 4	D_4 5	D_5 6	Total Req.
O_1 - 1	[1] 10	[2] 20	[3]	[4]	[5]	30
O_2 0	[2]	[3] 0	[4] 25	[5] 15 +	[1]	40
O_3 - 4	[3]	[4]	[5]	[1] 20 -	[2] 30 +	50
Total Avail.	10	20	25	35	30	120

Table 22[6]

	2	3	4	0	1	
O_1 - 1	[1] 10	[2] 20	[3]	[4]	[5]	30
O_2 0	[2]	[3] 0	[4] 25	[5]	[1] 15	40
O_3 1	[3]	[4]	[5]	[1] 35	[2] 15	50
	10	20	25	35	30	120

Table 23[6]

[0]	[0]	[0]	[0]	[5]	
[0]	[0]	[0]	[0]	[0]	
[0]	[0]	[0]	[-5]	[0]	

Example of the Assignment Method

The assignment method[6] is a special case of the transportation method and is illustrated in a problem of assigning four men to four jobs. It is assumed that each man is capable, to some extent, to perform each job. Furthermore, it is assumed that the degree of competence can be expressed by some numeric value, such as a score. The men and corresponding job competence in each job is indicated in Table 24. The objective is a balanced job assignment; it is desired to obtain a maximum total score over all jobs, where a high score indicates competence. The method may be used to maximize or minimize as in the simplex method.

The method consists in "shipping" fractions of men to the various jobs; mathematically the problem is:[6]

Maximize (or minimize) $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij}$

Subject to $\sum_{i=1}^{m} x_{ij}$ = total man required on the j^{th} job = 1 man

$\sum_{j=1}^{n} x_{ij}$ = total amount of i^{th} man assigned = 1 man

x_{ij} = amount of i^{th} man assigned to the j^{th} job.

c_{ij} = score of i^{th} man on j^{th} job.

$x_{ij} \geq 0$

The assignment method is possible because of the following theorem:

Theorem:[6] In the transportation problem when marginal data are integers, every basic solution comes out in integers.

In view of the above, x_{ij} must be 0 or 1 in each basic solution of the problem. To minimize total score, Table 24 may be used directly; to

maximize the total score, the scores must be changed to negative quantities. Thus maximizing the original score is equivalent to minimizing the negative of the score.

To maximize total score, Table 25 is obtained by changing the scores of Table 24 to negative quantities. Table 26 is obtained by subtracting the lowest number (- 97), from every number in Table 25, resulting in all positive scores.

For each iteration process, draw the minimum number of horizontal or vertical lines which will cross out all zeros in the array. Look for the smallest number in the array which is not under any of the above lines. Subtract this number from all numbers not crossed out by the horizontal or vertical lines. The process is repeated until there is at least one zero in every row and column. This condition will occur when the minimum number of horizontal and vertical cross out lines is equal to the number of rows or columns. From these zeros, select the zero so that it is the only one in each row and column. The optimal program is given in Table 31 as marked by the asterisks: man 1 to job 4, man 2 to job 3, man 3 to 2, and man 4 to job 1, total score is equal to 339, (97 + 63 + 89 + 90).

The above solution requires 5 iterations from Table 26. By utilizing the matrix characteristics of the array of Table 26, the answer may be obtained in 1 iteration. Any number may be added or subtracted to an entire row or an entire column in a matrix. Therefore, Table 32 is obtained directly from Table 26 by subtracting 18 from row 2, 3 from row 3, and 7 from row 4. Table 33 is obtained by subtracting 5 from column 2; no further zeros can be obtained in this manner. The assignment method, minimum number of horizontal or vertical lines to cross

out all zeros, is applied, 16 subtracted from the remaining array, and the optimal program of Table 34 obtained. The latter is identical to Table 31.

The methods discussed work only for a square matrix. However, the procedure can be used when the original matrix is not square by use of fictitious values and corresponding high or low "penalty" values which permit solution as described in the transportation method. In addition, the numerical score of this problem could have been replaced by time per unit, cost per unit, quality scale per vendor, etc., depending upon the application.

Table 24

	J_1	J_2	J_3	J_4
M_1	31	34	73	97
M_2	55	50	63	79
M_3	37	89	94	32
M_4	90	57	42	37

Table 25

	J_1	J_2	J_3	J_4
	-31	-34	-73	-97
	-55	-50	-63	-79
	-37	-89	-94	-32
	-90	-57	-42	-37

Table 26

J_1	J_2	J_3	J_4
66	63	24	0
42	47	34	18
60	8	3	65
7	20	55	60

Table 27

M_1	66	63	24	0
M_2	39	44	31	15
M_3	57	5	0	62
M_4	4	17	52	57

Table 28

	66	63	24	0
	35	40	27	11
	57	5	0	62
	0	13	48	53

Table 29

	66	63	24	0
	24	29	16	0
	57	5	0	62
	0	13	48	53

Table 30

M_1	66	58	24	0
M_2	24	24	16	0
M_3	57	0	0	62
M_4	0	8	48	53

Table 31

	50	42	8	0*
	8	8	0*	0
	57	0*	0	62
	0*	8	48	53

Table 32

66	63	24	0
24	29	16	0
57	5	0	62
0	13	48	53

Table 33

M_1	66	58	24	0
M_2	24	24	16	0
M_3	57	0	0	62
M_4	0	8	48	53

Table 34

	50	42	8	0*
	8	8	0*	0
	57	0*	0	62
	0*	8	48	53

DUAL THEOREM

The "dual" problem, previously mentioned, is developed in references (1), (4) and (6). The geometric concept of the direct problem and its dual problem is illustrated for a two-dimensional problem in reference (21). The value of the dual problem is centered on the theorem that if a solution exists to either the direct problem or its dual, then a solution to the other also exists. In the notation previously used, the theorem states:

$$\max. \sum_j c_j x_j = \min. \sum_i w_i b_i$$

Thus, in formulating the model, it is possible to work with either the direct problem or its dual, and the solution of either automatically provides the solution of the other. The answer to the dual, the w_i's, may be read directly from the final tableau from the $z_j - c_j$ row in the columns associated with the initial slack vectors. The w_i, an "opportunity" value, in the dual problem provides information on the rate of change of the functional with changes to the capacities or constraints of the problem. There is no method of determining which form of the problem will yield the easiest solution. It is recommended that both forms be examined before attempting to solve either form.[6]

Use of the dual theorem is made in reference (22), a warehousing problem with multiple products and varying prices. It also illustrates special computational features possible due to the structure of that problem.

SUMMARY

Two important phases of linear programming are formulation of the mathematical model and the computational aspects.

The majority of linear program examples published to date involve one or more of the methods presented. It is to the advantage of management and industrial engineers to become acquainted with the basic methods and capabilities of mathematical programming, particularly in industries that do not maintain a staff versed in this field. By doing so, some of the necessary groundwork can be handled by present personnel to reduce the time and expense in utilizing consultants when required. Formulating the mathematical model requires personnel who are thoroughly qualified in the use of the various techniques, and this phase determines the success or failure of the whole process.

For a large problem, solution by computer is the only efficient means, not because of the complexity of the mathematics but due to the quantity involved. A transportation problem of 25 origins and 60 destinations required nine man days by hand computation techniques. An IBM card programmed calculator can solve a 20 by 20 matrix in about three hours. The Rand Corporation (23) simplex code can solve 100 equations in about five hours, and their code for a transportation type problem can solve a 300 equation system in about one hour.

Rental time at data processing centers, located in various parts of the country, offer one possibility for small companies unable to justify the large capital investment of a computer.

References (24) and (25) discuss methods of machine matrix inversion. The method of reference (24) is not well suited to desk calculator

application because of the amount of record keeping at each step. Reference (26) provides a complete description on setting up a problem for solution by an IBM 701 computer.

REFERENCES

(1) A. Charnes, W. W. Cooper, and A. Henderson, An Introduction to Linear Programming. New York, John Wiley and Sons, Inc., 1953.

(2) Reference (4), Chapter XXI.

(3) R. Dorfman, Application of Linear Programming to the Theory of the Firm, Berkeley, University of California Press, 1951.

(4) Activity Analysis of Production and Allocation (T. C. Koopmans, ed.) Cowles Commission for Research in Economics Monograph No. 13. New York, John Wiley and Sons, Inc., 1953.

(5) J. R. Jackson, Detailed Scheduling of Industrial Production Activities, Econometrica, Vol. 21, 1953.

(6) A. Charnes, unpublished notes in Mathematical Programming and Industrial Management, a course conducted by Professor A. Charnes at Purdue University, 1955.

(7) R. Bellman, Dynamic Programming, Mathematics for Modern Engineering, University Extension, University of California, Lecture XVII, ditto form.

(8) C. C. MacDuffee, The Theory of Matrices.

(9) Blackwell and Girshick, Theory of Games and Statistical Decisions. New York, John Wiley and Sons, Inc., 1954.

(10) Reference (4), Chapter XVII.

(11) Reference (4), Chapter XVIII.

(12) Reference (4), Chapter III.

(13) Proceedings of the Association for Computing Machinery, Richard Rimbach Associates, 1952.

(14) A. Charnes, Optimality and Degeneracy, Econometrica, Vol. 20, 1952.

(15) E. L. Arnoff, The Application of Linear Programming to Production Engineering and Scheduling, A.S.M.E., Paper No. 54-A-223.

(16) E. L. Arnoff, Short Course in Operations Research Notes, Case Institute of Technology, 1954.

(17) G. H. Symonds, Linear Programming: The Solution of Refinery Problems. Esso Standard Oil Company, New York, N. Y., 1955.

(18) Reference (4), Chapter XXIII.

(19) A. Henderson and R. Schlaifer, Mathematical Programming. Harvard Business Review, Vol. 32, No. 3, 1954.

(20) A. Charnes and W. W. Cooper, The Stepping-Stone Method of Explaining Linear Programming Calculations in Transportation Problems, Management Science, October, 1954.

(21) A. W. Tucker, Linear Programming, American Society for Quality Control, Abstracts of Papers, 1953.

(22) A. Charnes and W. W. Cooper, Generalization of the Warehousing Model, Operational Research Quarterly, Vol. 6, No. 4, December, 1955.

(23) G. B. Dantzig, Status of Solutions of Large Scale Linear Programming Problems, Rand RM 1375, November, 1954. The Rand Corporation, Santa Monica, California.

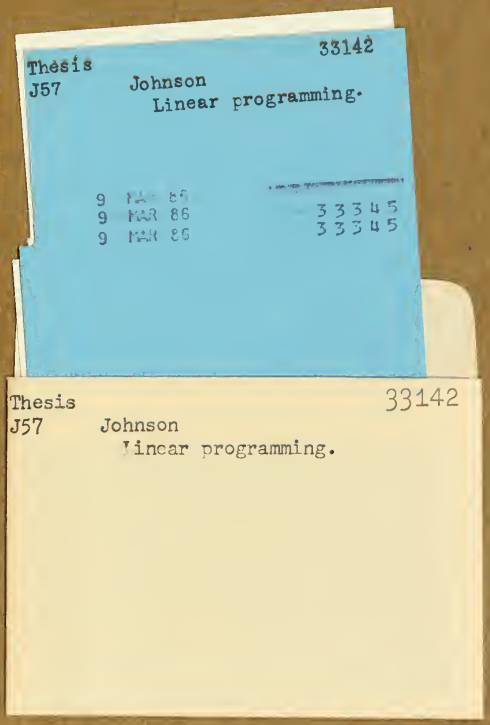
(24) H. M. Wagner, Matrix Inversion on an Automatic Calculator by Row and Column Partitioning, Rand P-417 of July, 1953. The Rand Corporation, Santa Monica, California.

(25) W. O. Hays, Computational Experience in Solving Linear Programming Problems, Rand P-482 of March, 1954. The Rand Corporation, Santa Monica, California.

(26) W. O. Hays, The Rand Code for the Simplex Method for the IBM 701 Electronic Computer, Rand RM 1440 of February, 1955. The Rand Corporation, Santa Monica, California.

(27) G. B. Dantzig, Linear Programming under Uncertainty, Management Science, April-July, 1955.

(28) A. Charnes, W. W. Cooper, and G. H. Symonds, Stochastic Programming of Heating Oil by the Horizon Method, ONR Report No. 33, Graduate School of Industrial Administration, Carnegie Institute of Technology.


Thesis 33142
J57 Johnson
Linear programming.

9 MAR 86 33345
9 MAR 86 33345
9 MAR 86

Thesis 33142
J57 Johnson
Linear programming.

thesJ57
Linear programming.
3 2768 001 02764 2
DUDLEY KNOX LIBRARY